An interactive evolution strategy based deep convolutional generative adversarial network for 2D video game level procedural content generation.

JIANG, M. and ZHANG, L.

2021

An Interactive Evolution Strategy based Deep Convolutional Generative Adversarial Network for 2D Video Game Level Procedural Content Generation

Ming Jiang
School of Computer Science
University of Sunderland
Sunderland, United Kingdom
Ming.Jiang@sunderland.ac.uk

Li Zhang
National Subsea Centre
Robert Gordon University
Aberdeen,Unitied Kindom
L.Zhang7@rgu.ac.uk

***Abstract*—The generation of desirable video game contents has been a challenge of games level design and production. In this research, we propose a game player flow experience driven interactive latent variable evolution strategy incorporated with a Deep Convolutional Generative Adversarial Network (DCGAN) for undertaking game content generation with respect to a 2D Super Mario video game. Since the Generative Adversarial Network (GAN) models tend to capture the high-level style of the input images by learning the latent vectors, they are used to generate game scenarios and context images in this research. However, as GANs employ arbitrary inputs for game image generation without taking specific features into account, they generate game level images in an incoherent manner without the specific playable game level properties, such as a broken pipe in the Mario game level image. In order to overcome such drawbacks, we propose a game player flow experience driven optimised mechanism with human intervention, to guide the game level content generation process so that only plausible and even enjoyable images will be generated as the candidates for the final game design and production.**

Keywords—Interactive Evolution Strategy, Deep Convolutional Generative Adversarial Network, Procedural Content Generation, Video Game

I. INTRODUCTION

A. Procedural Content Generation (PCG)

Procedural Content Generation (PCG) is algorithmic design and creation of game contents, such as game levels, weapons, obstacles, and characters. It is conducted by an automatic programme during the game design process, with limited or indirect input from game designers [1]. On one hand, PCG can help free up game designers from tedious manual creation of games contents; on the other hand, this algorithmic approach is able to facilitate a so-called mixed-initiative design paradigm of a human designer or player cooperating with the generative algorithm to generate the desired game contents, which can be both more computationally creative and better tailored for the game players. A PCG system refers to a process that incorporates a PCG method as one of its components [2]. As an example, through measuring and using Neural Networks of an AI-assisted game design tool to model the response of players in a novel game level generation scenario, the PCG system can create player-adaptive games in a generate-and-test loop until a satisfactory solution is obtained, so that it maximises the enjoyment and playable factors of the game. As suggested in [2], common desirable properties of a PCG solution are its Speed, Reliability, Controllability, Expressivity, Diversity, Creativity and Believability. In this research, we are more focused on fulfilling the Expressivity, Diversity, Creativity and Believability prosperities, which are more related to the Computational Creativity aspects of the proposed deep learning and interactive evolutionary computation based PCG system.

In recent years, Convolutional Neural Network (CNN), Generative Adversarial Network (GAN), and Autoencoders are the popular Deep Neural Networks that have been used as part of a PCG system to generate games level images. A CNN model can be trained to extract the latent features from the training datasets, for example, from a set of 2D game level images. Then a GAN model is trained to generate creative new game levels from these latent features by interpolations or extrapolations in the latent dimensions. An Autoencoder can be trained to encode the original high dimensional training games level images into low dimensional latent features first and then reconstruct a new high dimensional image from the latent features [3] [4][5].

B. Deep Convolutional Generative Adversarial Network (DCGAN)

The GAN model [6] includes two competitive components: the generative neural network mapping sampled latent variables to data space and the discriminator neural network assigning a detection probability of a real training sample or a generated fake instance by the generative network. The objective of the GAN model is to train the generator to generate perfect fake instances to fool the discriminator and simultaneously to train the discriminator to detect these fake instances from the actual data as accurately as possible. The sampled latent variables for fake instances generation are normally chosen independently by the generator from a random distribution, such as Gaussian noise vectors. Since the discriminator is trained in a manner of supervised learning with actual data that bear the ground truth, the criticism by the

discriminator can force the generator to generate higher quality instances. With respect to the training of the discriminator in the GAN model, a series of CNNs but without pooling layers can be employed to automate the challenging feature extraction process. Once the GAN model is trained, the discriminator will be discarded and only the generator will be kept to generate new data instances that reflect the essential characterises of the training examples. The inputs to the trained generator are normally fixed size latent variables that are randomly sampled from a space of Gaussian noise vectors. Besides the standard GAN model, [7] proposes a Deep Convolutional Generative Adversarial Network (DCGAN) architecture, which extracts the features of image representations with strided convolution rather than the conventional dense network used in the original GAN model.

C. Interactive Latent Variable Evolution (LVE) Strategy

Whilst those randomly and independently chosen samples of latent variables would be used to generate new data instances, a well devised search strategy of sampled latent variables could be used to generate instances with certain desirable features. For selecting salient input latent variables for the generator, Latent Variable Evolution (LVE) approaches [8][9] have been proposed for searching the most suitable latent variable for the GAN model to generate an image that is closely matched to the targeted/trained images.

Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES)[10] is a well-known search strategy for evolving vectors of real numbers and is efficient for optimizing non-linear non-convex problems in the continuous domain without a-priori domain knowledge. The optimization process of CMA-ES does not rely on the assumption of a smooth fitness landscape. CMS-ES has been applied into the evolution of the latent vectors for Mario game level image generation [3]. The effectiveness of the evolution strategy can be improved with an interactive user guide and directed with the evaluation feedback from the image user during the evolution. A user's evaluation feedback can be incorporated into the fitness function for guiding the evolution of the latent variables. For example, the best generated images are selected by the user. Then the latent variables that produced these images are evolved to create a set of new much desirable variables by using a crossover operation [9].

II. The Proposed PCG Method

A. A Flow Experience Driven Interactive LVE Strategy

In this research, we propose a flow experience driven interactive evolutionary strategy to extend the CMA-ES based DCGAN for Mario game level generation [3]. In particular, we take a game player's self-reported flow experience [11] as the game level playability evaluation feedback and incorporate it interactively and periodically into the latent variable fitness function for the LVE strategy. The feedback signal is used as an additional critic term to the fitness evaluation function on the game level validated by the play through of the game simulation agent.

In the most general sense, flow is an optimal experience state in which people report feelings of concentration, deep enjoyment and complete absorption in an activity. The game contents that are generated and presented to the game players affect the perceived flow/optimal experience while playing the game. From the perspective of game player's experience, the Flow Channel Theory [12] is arguably the most important concept for game design. A Flow Channel is where the player's skills and game's challenges are balanced effectively, and the player is neither anxious nor bored. Outside the channel, the player is too anxious, or too bored, it will drive them to frustration. Frustrated players are very likely not to continue playing the game.

We extend the conventional game playthrough statistics data-based fitness evaluation at a game level to incorporate a player's self-reported flow experience information. In other words, we evaluate the overall fitness of a game level with a new objective function as follows.

$$O=P+J+FS \quad (1)$$

where, P denotes the fraction of the level that is completed by the game simulator agent in terms of progress on the x-axis, with a value range of 0-1,

J represents the number of jump actions performed by the game simulator agent, which is a non-deterministic value,

FS indicates the player's self-reported flow experience, with a value range of 0%-100%, on a particular periodical level generation.

The proposed LVE strategy aims to maximise the objective score function O of latent vectors during the evolution process, in which an optimised vector sample is deliberately selected to be mapped onto games level image space. The role of the FS factor is to induce a more stable and much faster convergence of the objective function by adding into human player's interactive critic (e.g. positive reinforcement). The degree of convergence acceleration will be influenced by the value of FS.

It is worth noting that the value of P, J, and FS can be all generated by a game player for each game level under evaluation. However, under such contexts, the whole evaluation process will be conducted by a human game player rather than mainly by the game simulator agent, which can be quite a tedious task. Therefore, in this research, we alleviate such problems by combining evaluation with human and game simulator agent together to balancing the human intervention and agent-based automation using GANs.

B. 2D Super Mario Game Level Image Generation Procedure

The generated game level images/contents will be played in the game by the player to guide the further evaluation of the latent vector space in return. The CMA-ES method is extended with the proposed fitness function to realise the player directed LVE strategy, which follows 7 steps:

1. Set up Wasserstein GAN (WGAN) process to train the generator and discriminator.
2. The trained generator produces a set of games level contents from input latent vectors .
3. Set up the Mario game engine to stage the generated game level.

4. Player interactively assigns a satisfaction score for a latent vector based on his/her self-reported flow experience in playing the corresponding game level with the Mario game engine.
5. The Mario game engine calculates the fitness score for the latent vector with parameters, including the satisfaction score, defined in the objective function.
6. CMA-ES evaluates the fitness score of latent vectors to direct the further evolution of latent vectors.
7. The Mario game engine stages an improved game level from an improved latent vector produced by the previous iteration of the evolving process.

III. The Proposed PCG System

The proposed PCG system shown in Figure 1 consists of a DCGAN, a CMA-ES based game level evaluator, and a Mario game AI simulator. Since the game level evaluation strategy is human in the loop based, a game level designer needs to play the Mario game simulator to provide the part of the subjective evaluation feedback to the CMA-ES based game level evaluator during the PCG procedure. The Mario game level representation is encoded as tiles according to the Video Game Level Corpus (VGLC) [5].

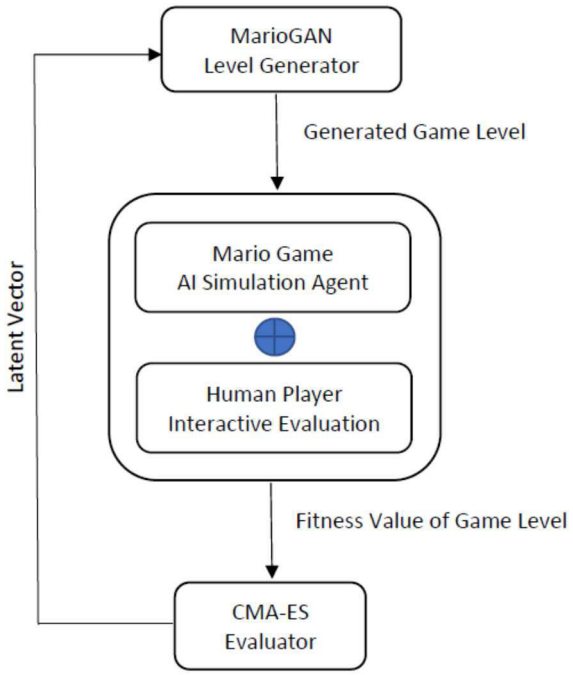


Fig. 1. The architecture & sataflow of the PCG system

A. Mario Game Level Representation

The Mario game level is built with a set of basic tiles (e.g., solid/ground, enemy and pipes) whose properties are represented by the particular character symbols of the Video Game Level Corpus (VGLC) [5]. Each VGLC character symbol is mapped into a distinct integer, which is converted to a one-hot encoded vector as input into the discriminator of MarioGAN. The generator model of MarioGAN also represents its outputs in a one-hot encoded format, which are converted back to a collection of integer values. These integer values are sent to the Mario AI simulation agent for rendering and playing through. The detailed mapping from VGLC tile types and symbols, to MarioGAN training number codes, and finally to Mario AI simulation agent for tile visualizations are illustrated in [3].

B. MarioGAN

MarioGAN [1] is a Pytorch implementation of the GAN model that interprets the structure of Super Mario Bros games levels. MarioGAN is based on a DCGAN model that adapts from the model in [3] and trained with the WGAN [13] [2] algorithm. The MarioGAN network uses strided convolutions in the discriminator and fractional-strided convolutions in the generator. Batchnorm processing is applied in the generator and discriminator after each layer. The generative model of MarioGAN is trained on actual Mario levels from the Video Game Level Corpus for generating new level segments from latent vectors, and these segments can be stitched together to make complete levels. As demonstrated in [3], the MarioGAN model with a promising LVE strategy is able to generate playable game levels. The strategy can be either purely based on the static properties of the generated level (e.g., certain tile distributions) or on the results of simulations of playing the level by an artificial agent. The trained generative model evolves game levels with a Java version of CMA-ES and by playing levels with Robin Baumgarten's A* Agent.

In our work, we extend the 'Simulations of Game Evaluation' function in [3] to incorporate a human game player's interactive evaluation with the simulation agent's evaluation in parallel. With the extension, the trained generator is effectively to generate game levels under an ensembled evaluation of the game simulation agent playing through the level and the human game player's subjective flow score for the corresponding game level.

C. CMA-ES Evaluator

The CMA-ES method [10] is widely used for optimizing non-linear non-convex problems in the continuous domain without a-priori domain knowledge. It does not rely on the assumption of a smooth fitness landscape. MarioGAN uses a Java based CMA-ES [3] to evolve the latent vector of real numbers and apply a fitness function of calculating simulation agent's level playing through performance. Under the guidance of a particular fitness function, CMA-ES searches through the space of latent vectors to produce levels that pass the evaluation of the games simulation agent as well as the satisfaction of the flow experience of human player.

D. Mario Game AI Simulation Agent

The A* agent by Robin Baumgarten[4] performs at a super-human level to determine the playability of a given level. Given jumping is the main mechanic in Mario and is required to overcome obstacles such as holes and enemies, the correlation between the number of jumps and difficulty is a

1 https://github.com/TheHedgeify/DagstuhlGAN

2 https://github.com/martinarjovsky/WassersteinGAN

3 http://cma.gforge.inria.fr/

4 https://www.youtube.com/watch?v=DlkMs4ZHHr8

reasonable assumption for playing the Mario game [3]. The number of jump actions performed by the agent can realistically approximate the experienced difficulty of playing through the game level as if being played by a human player.

E. Human Player's Interactive Evaluation

The human player interactive evaluation employs a human in the loop to perceive flow experience-based extension in the evaluation strategy for CMA-ES to accelerate the search of optimised latent vectors. Owing to the incorporation of human evaluation intervention, the revised fitness function is more capable of producing specific and plausible game levels effectively in comparison with those generated using the original fitness function. We introduce the evaluation details below.

IV. Experiments and Discussion

A. Experiment Configuration

We adopt experiment configuration settings in [3] for our experiments.

1) Level for Training MarioGAN

The MarioGAN method is trained with a level file from the VGLC encoded original Nintendo game Super Mario Bros [3]. Each input training image was generated by sliding a 28 (wide) x 14 (high) window over the raw level from left to right, one tile at a time. The width of 28 tiles is equivalent to the width of the screen in Mario. 173 training images are cropped from the level file in total. The size of the latent vector input to the MarioGAN generator is 32. In total 10 types of tiles with each integer tile are expanded to a one-hot vector. The training inputs for the discriminator are 10 channels. i.e. one-hot across 10 possible tile types.

2) Parameters of CMA-ES Evaluator

The CMA-ES process runs 20 iterations to generate 2240 levels in total. During each iteration, 112 latent vectors are generated to identify the best candidate from the evolution population. The maximum number of searching samples for finding one best candidate is 100.

3) Parameters of Human Player's Interactive Evaluation

The human player's interactive evaluation is scheduled at the end of each iteration, i.e. for every 112 generated levels, a game player perceived flow experience score is added into the latent vector/level fitness value, which is calculated by the objective function introduced in Section II.A with the simulation agent's performance of playing through the level.

B. Experimental Results

The purpose of the experiments is to demonstrate the pattern of how the human player's interactive evaluation would help stabilise and accelerate convergence of the objective function with the simulation agent together.

Figure 2 illustrates the bumping convergence of the mean fitness values of the generated game level over 20 iterations. Figure 3 shows a convergence acceleration of the average fitness values with human player's interactive evaluation with a flow score included at the end of each iteration over the 20 iterations. At the iteration 9, the first convergence emerges from iteration 9 to 12. Figure 4 demonstrates the human player's perceived flow score for each iteration. There is a positive correlation between the human player's interactive evaluation and the generation of game level with better fitness. Figure 5-7 are the similar correlation patterns around iterations 6 and 36 of another 50-iteration evaluations. Figure 8 shows the evolution of generated game levels over a 20-iteration evaluation. These game levels are gradually fit into the expectation of a human player.

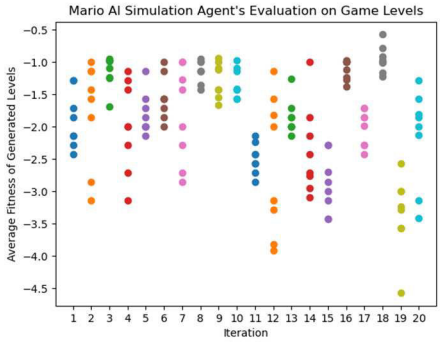


Fig. 2. Convergence of the mean fitness values of generated game levels during 20 Iterations (Lower values are better in CMA-ES.)

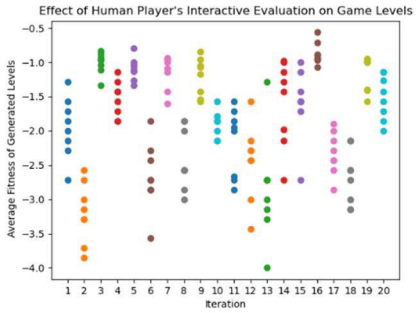


Fig. 3. Convergence acceleration of the flow score included mean fitness values of during 20 iterations (Lower values are better in CMA-ES.)

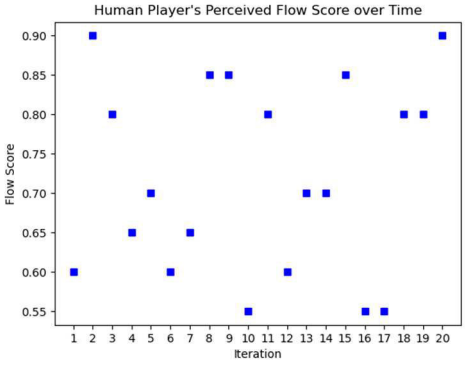


Fig. 4. Human player's perceived flow score at the end of eachiteration during 20 Iterations (The higher values are better.)

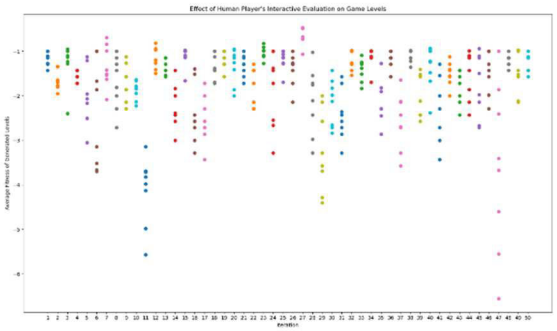


Fig. 5. Convergence of of the mean fitness values of generated game levels during 50 iterations (Lower values are better in CMA-ES.)

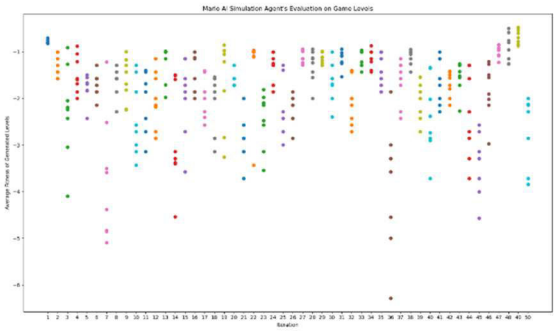


Fig. 6. Convergence acceleration of the flow score included mean fitness values of during 50 iterations (Lower values are better in CMA-ES.)

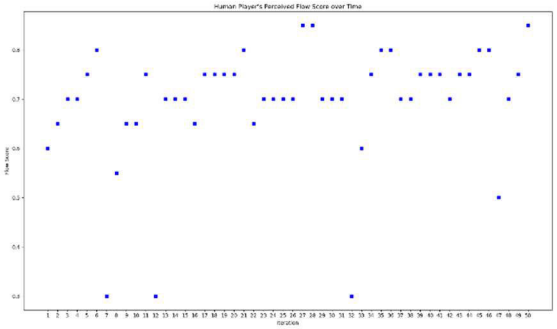
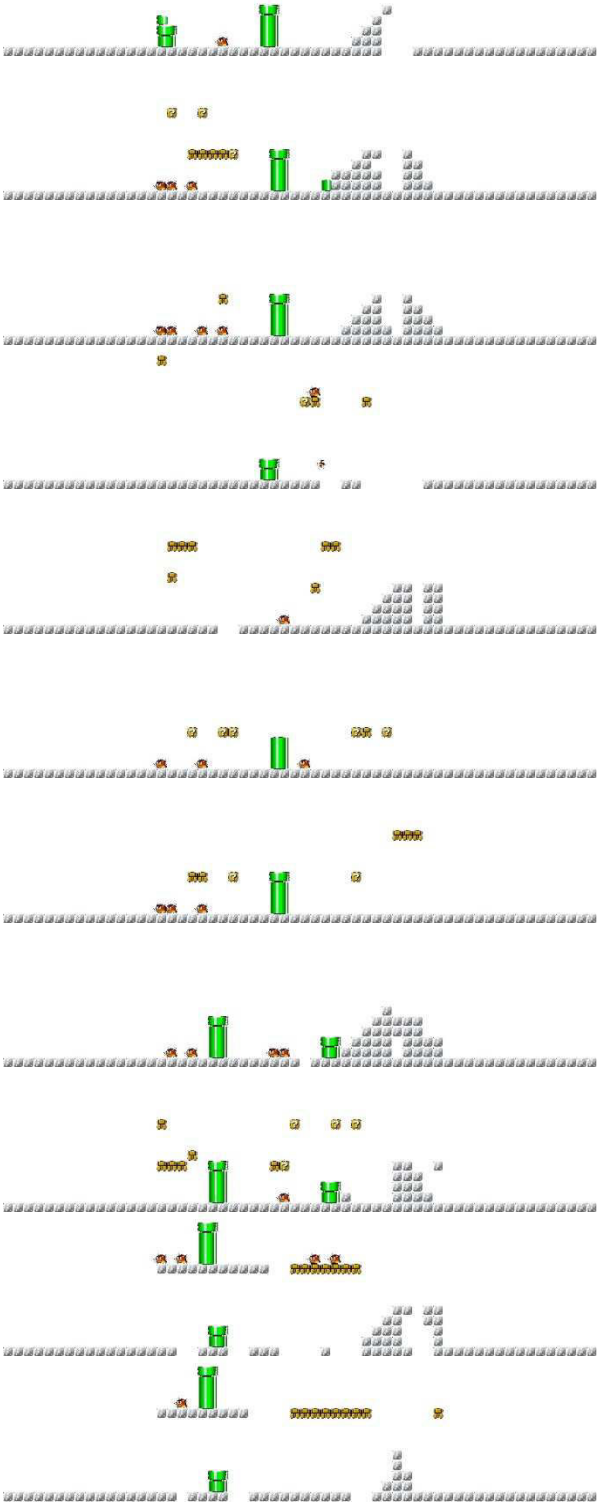


Fig. 7. Human player's perceived flow score at the end of each iteration during 50 iterations (The higher values are better.)

C. *Discussion*

The experimental results demonstrate the stabilisation and acceleration of the convergence of the latent vector's optimised objective score are correlated with positive enforcement of a player's flow score. While this is an effective pattern from the evolution perspective, there is also subjective nature of human interventions from the game level design perspective. The good balance between effective level generation and objective human intervention shall be a critical consideration as part of the whole strategy of game level design and production, for example, to assign different weights/frequency to human interventions and agent-based evaluations.



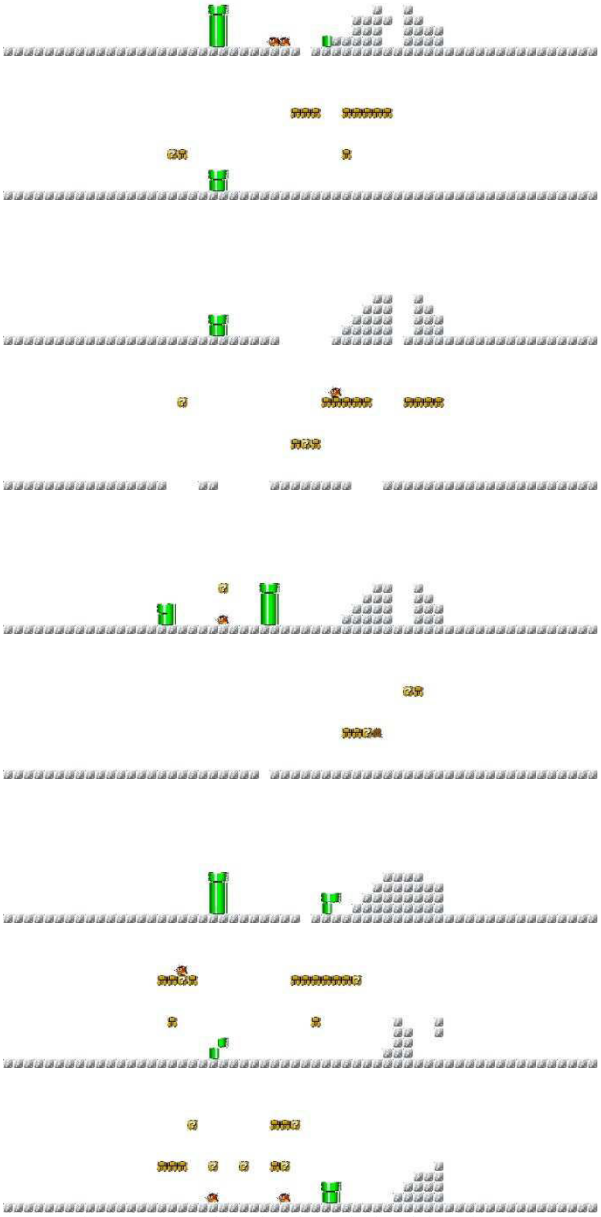


Fig. 8. Generated level at the end of each iteration from 1 to 20

V. CONCLUSION

This research presents a human player's flow experience driven interactive latent variable evolution strategy for guiding a DCGAN model to algorithmically generate desirable 2D Super Mario video game level contents. The evolution strategy is incorporated into and evaluated with game simulation agent's performance-based evolution strategy as a positive reinforcement to stabilise and accelerate the convergence of the mean fitness scores of generated game levels. In the future, the effectiveness of the interactive strategy will be further investigated with the trade-off between intervention frequency and fitness improvements of the generated game levels. It is also interesting to train the simulation agent directly with reinforcement learning or imitation learning mechanisms with hyperparameter fine-tuning by swarm intelligence algorithms [14][15][16][17][18] to learn the implied human flow experience in advance, so that it is possible to minimise the frequency of human intervention further during the level design generation process.

REFERENCES

[1] J. Togelius, E. Kastbjerg, D. Schedl, and G. N. Yannakakis, "What is procedural content generation? Mario on the borderline", *Proc. the 2nd Workshop on Procedural Content Generation in Games*, 2011

[2] J. Togelius, N. Shaker, and M. J. Nelson, "Procedural Content Generation in Games: A Textbook and an Overview of Current Research", Springer, ISBN 978-3-319-42714-0, 2016.

[3] V. Volz, J. Schrum, J. Liu, S. M. Lucas, A. Smith, and S. Risi, "Evolving Mario levels in the latent space of a deep convolutional generative adversarial network", *Proc. Genetic Evol. Comput. Conf.*, pp. 221-228, 2018.

[4] R. Jain, A. Isaksen, C. Holmgård, and J. Togelius, "Autoencoders for Level Generation, Repair, and Recognition", *Proc. ICCC Workshop on Computational Creativity and Games*, arXiv preprint arXiv:1702.00539 2016.

[5] A. J. Summerville, S. Snodgrass, M. Mateas, and S. O. Villar, "The VGLC: The Video Game Level Corpus", *Proc. the 7th Workshop on Procedural Content Generation*, 2016.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, "Generative Adversarial Nets". *Proc. Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

[7] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks", *CoRR*, abs/1511.06434, 2015.

[8] P. Bontrager, J. Togelius, and N. Memon, "DeepMasterPrint: Generating Fingerprints for Presentation Attacks", *arXiv preprint arXiv:1705.07386*, 2017.

[9] P. Bontrager, W. Lin, J. Togelius, and S. Risi, "Deep Interactive Evolution" *Proc. International Conference on Computational Intelligence in Music, Sound, Art and Design: EvoMUSART Springer,* pp. 267-282., 2018.

[10] N. Hansen, S. D Müller, and P. Koumoutsakos, "Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)", *Evolutionary Computation* 11(1), pp. 1–18, 2003.

[11] M. Csikszentmihalyi, *Flow: The Psychology of Happiness*, Rider, 1992.

[12] J. Schell, *The Art of Game Design*; *A Book of Lenses*, *3rd Edition*, CRC Press, 2020.

[13] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein Generative Adversarial Networks". *Proc. of the 34th International Conference on Machine Learning*, *PMLR* (70), pp. 214-223, 2017.

[14] W. Srisukkham, L. Zhang, S.C. Neoh, S. Todryk and C.P. Lim, "Intelligent Leukaemia Diagnosis with Bare-Bones PSO based Feature Optimization," *Applied Soft Computing*, 56. pp. 405-419. 2017.

[15] K. Mistry, L. Zhang, S.C. Neoh, C.P. Lim, and B. Fielding, "A micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition," *IEEE Transactions on Cybernetics*. 47 (6) 1496–1509. 2017.

[16] T. Tan, L. Zhang and C.P. Lim, "Adaptive melanoma diagnosis using evolving clustering, ensemble and deep neural networks," *Knowledge-Based Systems*. 2019.

[17] T. Tan, L. Zhang, C.P. Lim. B. Fielding, Y. Yu, and E. Anderson, "Evolving Ensemble Models for Image Segmentation Using Enhanced Particle Swarm Optimization," *IEEE Access*. 2019.

[18] T. Tan, L. Zhang and C.P. Lim, "Intelligent skin cancer diagnosis using improved particle swarm optimization and deep learning models," *Applied Soft Computing*, p.105725. 2019.